

REMARKS

This Amendment is in response to the Office Action dated July 11, 2008 in which claims 1-22 were rejected. Applicant respectfully requests reconsideration and allowance of all pending claims in view of the above-amendments and the following remarks.

I. INFORMATION DISCLOSURE STATEMENT

Enclosed is an information disclosure statement containing two additional references obtained during prosecution of a corresponding Japanese patent application. Applicants respectfully request consideration of the references.

II. TELEPHONE INTERVIEW

Applicant would like to thank the Examiner for the courtesies extended during a telephone interview held on October 2, 2008 with Applicant's attorney, David Brush, and the Examiner's supervisor, Brian Pendleton. Applicant's attorney proposed an amendment to claim 1 to further distinguish Applicant's "pointer" with the "sequence numbers" of Okura. While no specific agreement was reached regarding patentability of the claims, Applicant's attorney agreed to submit a response for further consideration by the Examiner. The Examiner also suggested Applicant show support in the specification for newly added claim limitations.

III. REVIEW OF THE PRESENT APPLICATION

The present application relates to the data transmission in the form of data stream(s). Each data stream is made up of elementary stream units (or packets). An embodiment of the present application optimizes processing of these stream units when they are dependent of preceding stream units in the same stream, or in another stream.

In known techniques, an important difficulty is that of synchronization when the transmission is made in an asynchronous way. In this case, some stream units emitted after can be received before previously emitted ones. In such a case, one cannot process a received unit of stream, if it is dependent on a preceding unit of stream not yet received.

Specific protocols were thus developed to synchronize streams. That can however prove to be insufficient, or at the very least insufficiently efficient. Indeed, even in the presence of synchronized streams, a receiving terminal can miss a stream unit (which is used for example as

a basis for a sequence of images, or which contains a decoding key), making it impossible to process the whole stream.

The present application thus proposes a new and inventive approach, including synchronizing the elementary stream units. The streams themselves are not synchronized by the “pointers” of the claimed invention. Any synchronization of the streams themselves is realized by the “Object Descriptor” of a stream, which is dedicated to the description of the scene.

Furthermore, the “pointers” of the claimed invention provide a logical system of synchronization, or linking of the stream units (different and complementary of a simple numbering of the stream units), allowing management of the stream units, to limit the processing in terminals, and to improve quality of restitution.

According to the present claims, one thus provides, in at least some stream units, pointers, pointing towards (i.e. designating) at least another stream unit (of the same stream or of another stream) likely to be received previously, so that a treatment of the aforesaid stream unit is carried out only if the former unit were received.

The pointers thus provide, in other words, a “back” synchronization, carried out on the level of the stream units, and not, as in the prior art, carried out on the entire stream: the pointers allows precise synchronization with another stream unit to which the pointer points.

A. Examples From Specification

As shown in Figure 1, the processing of 14 base stream can begin before the processing of dependant stream 10: thus, the units of the base stream 14 are required by the dependent stream 10 but the units of the dependent stream are not required to process the base stream 14. Furthermore, in case of a transmission of multiple video streams, it is not necessary to have received the entire audio stream to begin the processing of the video streams. The fact that only the first stream unit of the audio stream has been received is sufficient (for the synchronization for example).

As shown in Figure 2, the top line illustrates a transmitted stream, the second line illustrates the stream received by a first receiver (session 1), and the third, bottom line illustrates the stream received by a first receiver (session 2).

Session 1 starts at packet 211, then takes into account the stream unit 213, to which the stream unit 211, then the stream unit 215 points (24), according to link 25.

Session 2 opens a little later in time, and starts with stream unit 212, which points (26) to stream unit 214.

We can see that a single stream (which is used in two sessions) contains the same pointers. The stream can be processed even if some non-required streams units have not been received:

- in session 1, the stream unit 212 has not been received,
- in session 2, the stream unit 213 has not been received.

Thus the processing, in the receiver terminal, has already began (session 1 and session 2 have started). But it will not be continued if stream unit 215 is not received.

Figure 2 is therefore a non-limiting example where a single stream is processed in two different manners. Stream unit 211 is transmitted first, but may arrive after stream unit 215. If 211 is a first required unit and 215 is a second required unit, the processing of the stream can begin when 211 arrives. Thus, units 212 and 213 may be used in two or more different manners, for example:

- a) They will simply be discarded as they arrived too late;
- b) unit 211 and unit 215 are used to initiate a rendering of a scene, and units 212, 213, 214 are use to populate some data in the scene.

B. Dependency Descriptor

As further described in the specification, such as on page 15, line 21 to page 16, line 26; page 18, lines 14-16 and page 35, line 27 to page 36, line 25, the pointer is a dependency pointer of length depLength. The dependency pointer is included in a dependency descriptor of the stream unit, where the dependency descriptor describes the dependency pointer.

What is described on page 36 is an example of the structure of a dependency descriptor (which is included in the stream unit). In this example, the dependencyLength is an extra variable in the dependency descriptor, and the length of the dependencyLength is 5 bits, for example.

The DependencyDescriptor may also include other fields such as:

- the mode field ;
- the ESID field ;

IV. PROPOSED CLAIM AMENDMENTS

Claim 1 (and similarly the other independent claims) is amended to clarify that:

said pointer is a dependency pointer of length depLength, the dependency pointer being included in a dependency descriptor of said stream unit, said dependency descriptor describing said dependency pointer.

In contrast, a simple “sequence number” is not described in a dependency descriptor.

In fact, a “sequence number does not, as require by claim 1, provide “at least one pointer that points to at least one stream unit of said stream or another stream that may have been received previously in a terminal.”

Basically, a sequence number allows a stream unit to say "I'm the third one". But the sequence number will not be able to say "I need the fifth stream unit of the second stream," for example, as the “pointer” is capable of designating in claim 1 of the present application.

We can say that the sequence number is the "name" of the stream unit, while the pointer of the invention is the name of an associated stream unit of the stream or of another stream, for example.

V. CLAIM REJECTIONS UNDER §101

Claim 17 was rejected as being directed to non-statutory subject matter, according to the current USPTO position.

With this amendment, claim 17 is amended into a method claim, which is statutory under §101.

VI. CLAIM REJECTIONS UNDER §102 and §103

Claims 1-22 were rejected as being allegedly anticipated by OKURA (U.S. Publ. 2001/0027468).

Claim 22 was rejected as being allegedly unpatentable over OKURA (U.S. Publ. 2001/0027468) in view of Herrmann (U.S. Patent No. 6,606,329).

A. **OKURA, U.S. 2001/0027468**

1. Summary

This document relates to the transmission of streams made of data packets. More precisely, this document relates to the dynamic management of the dependencies between several streams, which uses sequence numbers associated to packets. In other words, each packet has a number, and it should not be processed if previous packets, having a lower number, have not been received.

As already said, Okura does not disclose nor suggest the use of pointers and this is a major difference with the invention, but it is not the only one. Indeed, Okura does not disclose the use of such pointers for optimizing the decoding processing by controlling, previously, that the stream unit designated by the pointer has been received.

2. Demonstration of Novelty and Inventiveness

Pointers that are contained in an “Object Descriptor” cannot be compared to pointers of the invention. Indeed, pointers that are contained in Object Descriptors are used to build scenes. In other words, Object Descriptors allow mixing various streams (visual and audio for example) to build the scene to browse. In contrast, pointers in streams units, which are recited in claim 1 of the present application, do not have the same functions. These pointers differ from the sequence numbers disclosed by Okura.

Firstly, a pointer according to claim 1 points to another stream unit. The pointer of claim 1 does not point to an entire stream.

Consequently, the pointers of claim 1 are capable of synchronizing stream units with other stream units of the same stream or of another stream previously received.

Secondly, the pointers of claim 1 allow, for example, a stream to declare itself as dependent of another stream, stream unit-by-stream unit if needed. Such a dependency has no need to be declared in a centralized way by object descriptors. Therefore, such a dependency allows a rendering device (such as a video player) to take into account various streams, not only by using object descriptors (as in anterior arts) but with the use of the pointers of claim 1.

Thus, the difference between the pointers and the sequence numbers of Okura include:

1. The sequence number of Okura does not correspond to a dependency pointer.
2. The sequence number of Okura does not correspond to a dependency pointer having a length depLength.
3. The sequence number of Okura is not described in a dependency descriptor.
4. In the present application, the pointer in the current stream unit points to a specific previous required unit. In other words, the pointers are not required to be sequential.

5. With the pointer of claim 1, the procedure allows some stream units to be processed in a different order than transmitted. The pointers allow the current stream unit to be processed if the required previous unit or units (to which the pointer points) have been received, even if other, stream units of the stream that were transmitted between the current stream unit and the required previous unit have not yet been received. Okura does not disclose or suggest that the sequence numbers of Okura allow a current stream unit to be processed if other stream units of the stream have not yet been received.

6. The pointers point to a specific stream unit or units but does not point to all previous streams units.

7. The sequence number of Okura is the "name" of the stream unit, while the pointer of claim 1 is the name of an associated stream unit of the stream or of another stream, for example.

Therefore, the claimed invention is clearly new and non-obvious in view of this document. As a matter of fact, a simple numbering of units cannot be compared to the introduction of pointers into the stream units, because a sequence number only allows ordering the units and does not carry the necessary information permitting to organize a decoding process, and notably to verify that a specific, well identified, necessary stream unit (in the same stream or in another stream) was duly received.

Okura allows checking that all preceding stream units were received. But, an embodiment of the present invention allows checking that a specific, well identified, necessary stream unit was received (so that the current unit may be processed, even though some previous units were not received). This is not possible according to Okura.

Okura therefore does not anticipate the elements of claim 1, either in its previous form or in

its amended form.

B. Combination of Okura and Hermann

Since Okura does not anticipate the elements of claim 1, and since Hermann also does not disclose the pointers recited in claim 1, the combination of Okura and Hermann also does not teach or suggest the invention recited in claim 1 or the other independent claims.

The combination of Okura and Hermann is non-obvious, and also inefficient and inadequate. Indeed, combining Okura and Hermann would drive to:

- on one hand, numbering stream units; and
- on the other hand, a use of various processing methods depending on specific quality level and/or specifics options.

Thus, the simple combination of Okura and Hermann, we would have an architecture where the stream units are numbered consecutively and various processing methods would be used.

This does not lead to the invention recited in claim 1. To obtain the invention from Okura, it would have been necessary to:

- Detect the approach of Okura was not adequate to ensure decoding;
- Understand that it is desirable to take into account the distinct stream units, in order to simplify the processing of decoding;
- Decide to insert pointers in the stream units themselves to allow synchronizing stream units with other stream units of the same stream or another stream previously received.

None of these problems are described nor disclosed in Hermann.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /David D. Brush/

David D. Brush, Reg. No. 34,557
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319

DDB:

Phone: (612) 334-3222 Fax: (612) 334-3312